
Masters Theses

Student Theses and Dissertations

1972

Data compression design considerations for airborne test vehicles

Kenneth Carlyle Pittman

Follow this and additional works at: https://scholarsmine.mst.edu/masters_theses



Part of the [Electrical and Computer Engineering Commons](#)

Department:

Recommended Citation

Pittman, Kenneth Carlyle, "Data compression design considerations for airborne test vehicles" (1972). *Masters Theses*. 3556.

https://scholarsmine.mst.edu/masters_theses/3556

This thesis is brought to you by Scholars' Mine, a service of the Missouri S&T Library and Learning Resources. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

DATA COMPRESSION DESIGN
CONSIDERATIONS FOR AIRBORNE
TEST VEHICLES

BY

KENNETH CARLYLE PITTMAN, 1934-

A THESIS

Presented to the Faculty of the Graduate School of the

UNIVERSITY OF MISSOURI-ROLLA

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

1972

T2720
56 pages
c.1

Approved by

John A. Neill (Advisor) David D. Styer
S. J. Pagan

ABSTRACT

The design goals for a data compression system are developed using previous studies as a cornerstone. Sampling theory is touched on with qualitative coverage of its effects. Consideration is given to various methods of data compression with the advantages and disadvantages of each noted. The requirement for buffering is discussed along with various methods that may be employed to achieve specific buffering goals. The impact of data reconstruction and how it may affect the choice of compression techniques is briefly covered. Selected compression algorithms and ways of increasing system effectiveness through alterations to their basic structures are described in detail.

Implementation techniques are presented for a data compression system that can meet the previously developed design goals. The interrelations of various subsystems are described in quantitative terms. Constant emphasis is given throughout to ensure that the system is economical, practical and well within current hardware components and software techniques.

TABLE OF CONTENTS

	Page
ABSTRACT.....	ii
LIST OF ILLUSTRATIONS.....	v
I. INTRODUCTION.....	1
II. AVAILABLE METHODS OF DATA COMPRESSION.....	3
A. BACKGROUND.....	3
B. PREDICTORS.....	4
C. ZERO-ORDER PREDICTORS.....	4
D. INTERPOLATORS.....	5
E. ZERO-ORDER INTERPOLATORS.....	8
F. FIRST-ORDER INTERPOLATOR.....	8
G. RECONSTRUCTION CONSIDERATIONS.....	11
III. PROBLEM CONSIDERATIONS OF REMOTE TESTING.....	15
A. BACKGROUND.....	15
B. FUTURE PROJECTIONS.....	16
C. POSSIBLE SOLUTIONS.....	17
D. TIME AVERAGING.....	20
E. IDENTIFICATION.....	20
F. BUFFER CONSIDERATIONS.....	21
IV. DESIGN CONSIDERATIONS.....	23
A. ALGORITHM DESCRIPTIONS.....	23
B. HARDWARE COMPRESSOR VS. SOFTWARE COMPRESSOR.....	27
C. ALGORITHM MODIFICATIONS.....	29
D. USE OF A GENERAL PURPOSE COMPUTER FOR CONTROL.....	35

TABLE OF CONTENTS (continued)	Page
V. IMPLEMENTATION.....	37
A. BASIC COMPRESSOR.....	37
B. ANALOG SUBSYSTEM.....	39
C. LIST WORD.....	42
D. CONTROL WORD.....	42
E. INPUT INFORMATION.....	42
F. EDITING.....	44
G. CONTROL.....	44
H. DATA IDENTIFICATION.....	45
I. DATA TRANSMISSION.....	47
VI. CONCLUSIONS.....	48
BIBLIOGRAPHY.....	49
VITA.....	51

LIST OF ILLUSTRATIONS

Figures	Page
1. Zero-order Predictor Redundancy Reduction Method.....	6
2. Zero-order Predictor Reconstruction Technique.....	7
3. Zero-order Interpolator Redundancy Reduction Method.....	9
4. Zero-order Interpolator Reconstruction Technique.....	10
5. First-order Fan Redundancy Reduction Method.....	12
6. First-order Fan Reconstruction Technique.....	13
7. Compression Ratio Distribution.....	19
8. Flow Chart For Z.O.P.....	24
9. Flow Chart For Z.O.I.....	26
10. System Block Diagram.....	38
11. List Words.....	41
12. Control Words.....	43
13. Output Word Format.....	46

I. INTRODUCTION

More information about more aspects of various endeavors is being demanded by more people. Since many sources of information are remotely located from the centers that require the data, several forms of data transmission are employed.

If the volume of data is large, fairly constant and a short delivery time desired, a requirement is born for an economical method of collection and disbursement from the remote stations to the central facility. One obvious solution is the use of the existing telephone network. The major drawback to this choice is the cost of lines with sufficient bandwidth to transmit large volumes of data in a reasonable amount of time. The use of a voice grade line or less would be a good compromise if the data could be transmitted within the available bandwidth (1). Many industries use the telephone network for their communications in both analog and digital forms over a wide range of frequencies.

In some applications of data collection there are periods of oversampling. This results in large volumes of data for subsequent transmission. If this oversampled data were examined prior to transmission and only the significant data were transmitted, the resultant reduction in volume of data could facilitate employment of a line

with a narrower bandwidth. This removal of the non-significant or redundant data is often referred to as data compression.

One area of data collection which usually has the characteristic of being oversampled is that of instrumented airborne test vehicles. This is the particular area to which the research described in this paper was directed.

II. AVAILABLE METHODS OF DATA COMPRESSION

A. BACKGROUND

Data compression would be an impossible goal if optimum sampling were employed (2). Optimum sampling is the result of continuously adjusting the sampling rate of measurands to perfectly match their activity. This is not the case of most telemetry systems due to the high cost and complexity of adaptive sampling. Therefore, to adequately describe the periods of high activity, most measurands are oversampled. This oversampling has led to the development of several methods for data compression.

The use of special signal conditioners to extract specific information from a measurand is among the most generally used methods of data compression. This includes peak detectors, frequency analyzers and many other pre-transmission conditioning techniques. This particular form of compression is acceptable in some instances but is undesirable in many data systems. The undesirable feature is that the result cannot be used as an input to a reversing process to yield the original signal. The use of airborne compression techniques which utilize several inputs to calculate a final result for transmission is also subject to this same criticism.

A different type of airborne compressor which does not compute answers, but executes compression algorithms

within the test vehicle does not have this drawback. If the number of test vehicles is small and recovery is non-existent or unsure, then to include such an onboard system could be justified. When the number of test vehicles is large and the use of recoverable tape recordings is to be employed, the economic advantages of a minimum number of ground based data compression systems become overwhelming.

Compression techniques other than pre-transmission processing have been developed. Polynomial predictors and interpolators are among such methods that are practical and have been employed. The similarity between polynomials and telemetry data (3) has led to the acceptance of the polynomial as the most effective and widely used basis for compression algorithms.

B. PREDICTORS

Predictors are algorithms which predict that the next data point will be within some tolerance range of the last transmitted data point. If the prediction is correct the new data point is considered redundant and is discarded. These algorithms may be based on a polynomial of any order although the most common are those of zero and first order.

C. ZERO ORDER PREDICTORS

The predictor with the widest acceptance is zero-order with a floating aperture (3), (4), (5), (6), (7).

This algorithm predicts that the new data point is within some horizontal tolerance band of the last non-redundant data point. If the new point is not within this tolerance band it is considered non-redundant and transmitted. The new value for checking subsequent data is that of the most recently sent data with the tolerance band being imposed around it. Thus the tolerance band moves with the data creating a floating aperture. Figure 1 depicts this process. The reconstruction technique for this algorithm is shown in Figure 2. Using the first non-redundant point as the starting value, a horizontal line is drawn until the next non-redundant value is received. At this point a vertical line is drawn from the current value to the new value. The process is then repeated with another horizontal line continuing on from the new data value. This reconstruction technique is particularly suited to strip chart recorders. The displacement of the reconstructed data from the original sample points is never more than one half the tolerance band.

D. INTERPOLATORS

Interpolators are algorithms which allow all sample values between the last transmitted value and the present value to exert some influence on the transmitted values. The most commonly employed (3), (4), (6) polynomial interpolators are either zero-order or a first-order variation.

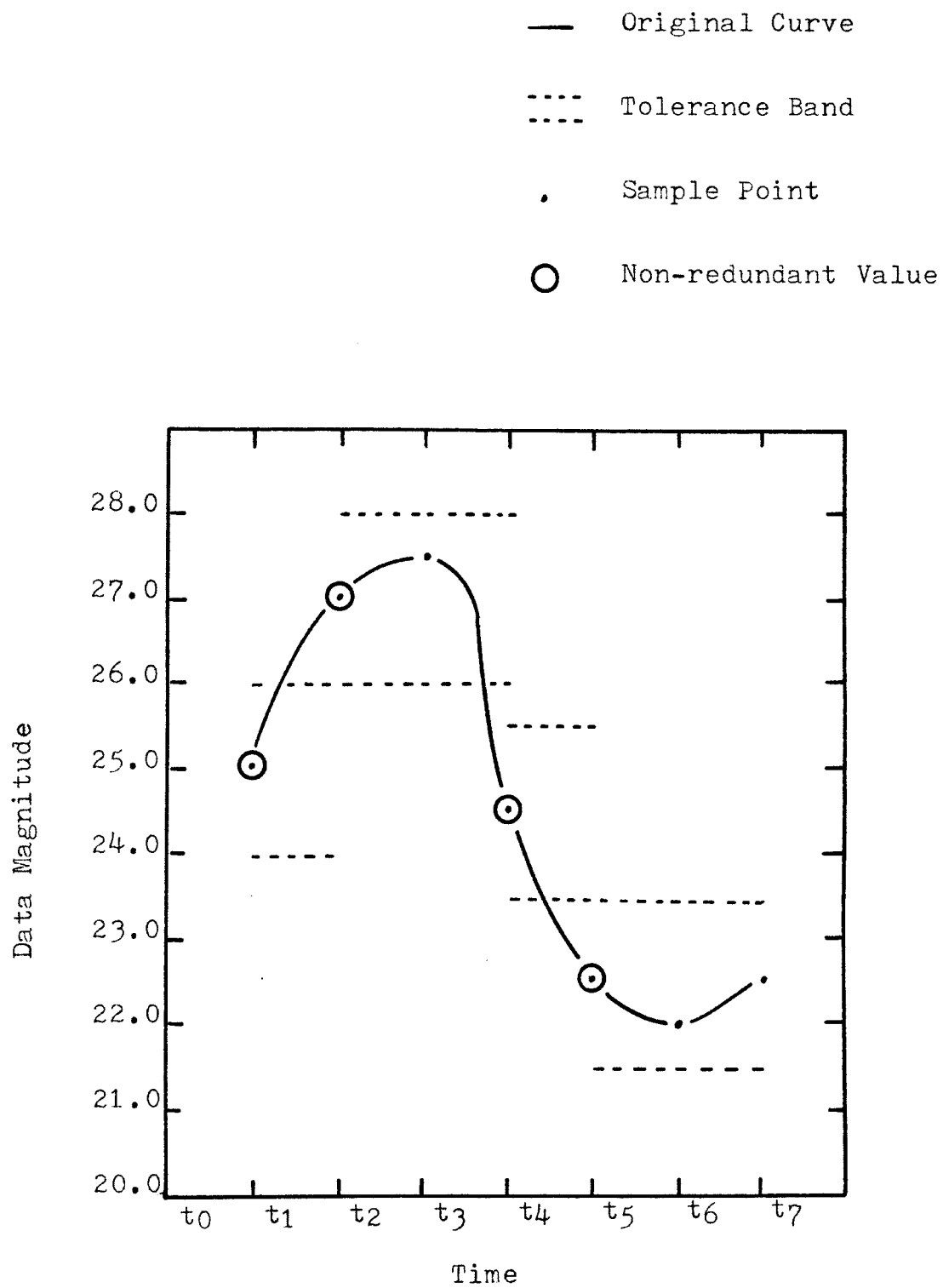


Figure 1. Zero-order Predictor Redundancy Reduction Method

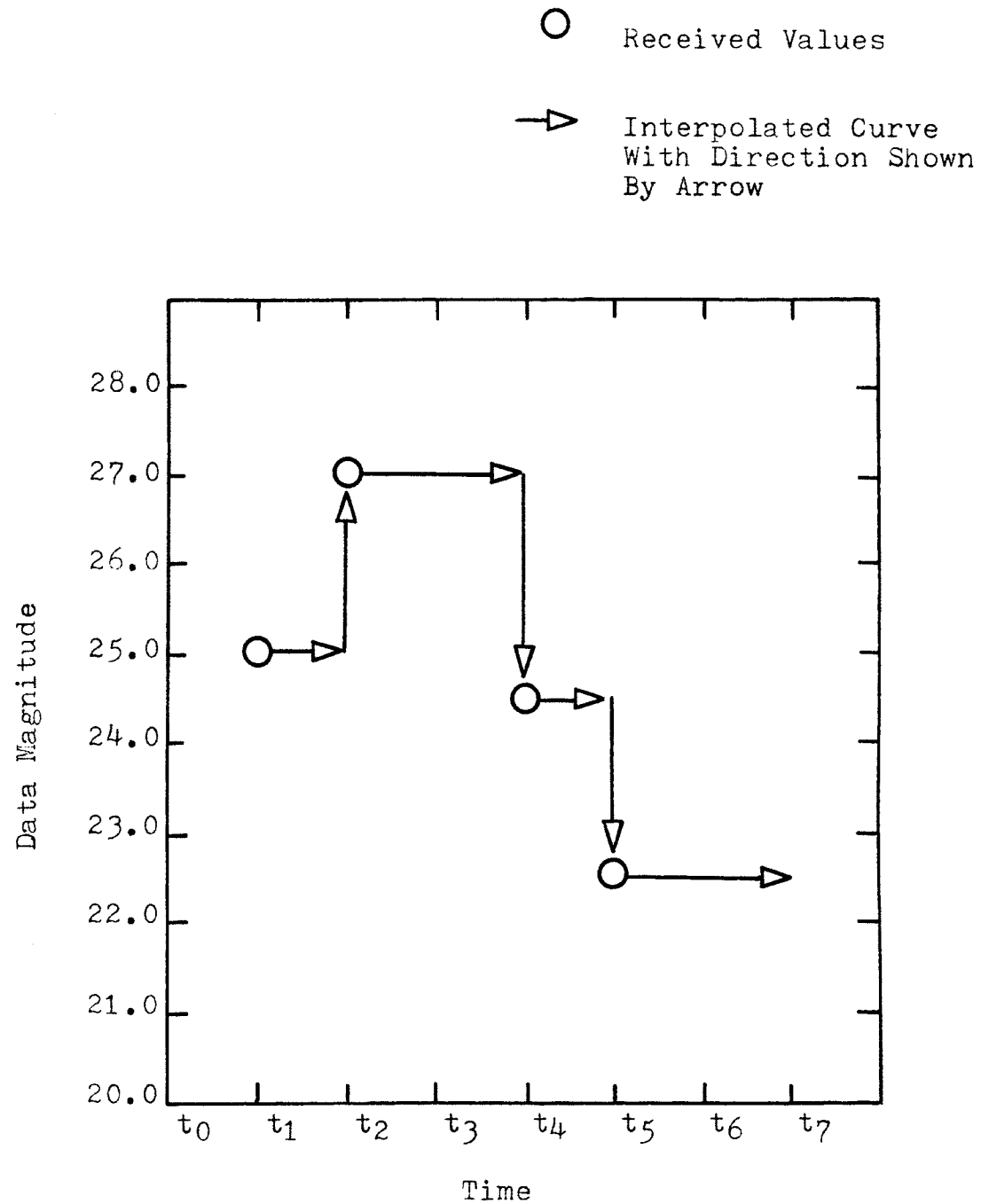


Figure 2. Zero-order Predictor Reconstruction Technique

E. ZERO-ORDER INTERPOLATORS

Zero-order interpolators, like the floating aperture, zero-order predictor, use a horizontal aperture. The basic distinction lies in the fact that the predictor uses only the value of the first sample to position its tolerance band, while the interpolator continues to adjust its tolerance band to encompass all possible samples within the specified tolerance band. Figure 3 depicts this process. The resultant transmitted value is the average value of the adjusted tolerance band based on all samples between the first and the sample point immediately preceding the sample which fell outside the tolerance band. A new initial tolerance band is then set up around the aforementioned sample, and the process is repeated. The reconstruction technique for this algorithm is shown in Figure 4. A horizontal line is drawn from the value at time t_{n+1} to t_n . Another horizontal line is drawn from the value at t_{n+2} to t_{n+1} with a vertical line drawn at t_{n+1} to connect these horizontal lines. The process continues in like manner as each point is received. The deviation from the original data is again never greater than one half the tolerance band at the sample points.

F. FIRST-ORDER INTERPOLATOR

The fan interpolator (4), (6), (8), (9) is a first-

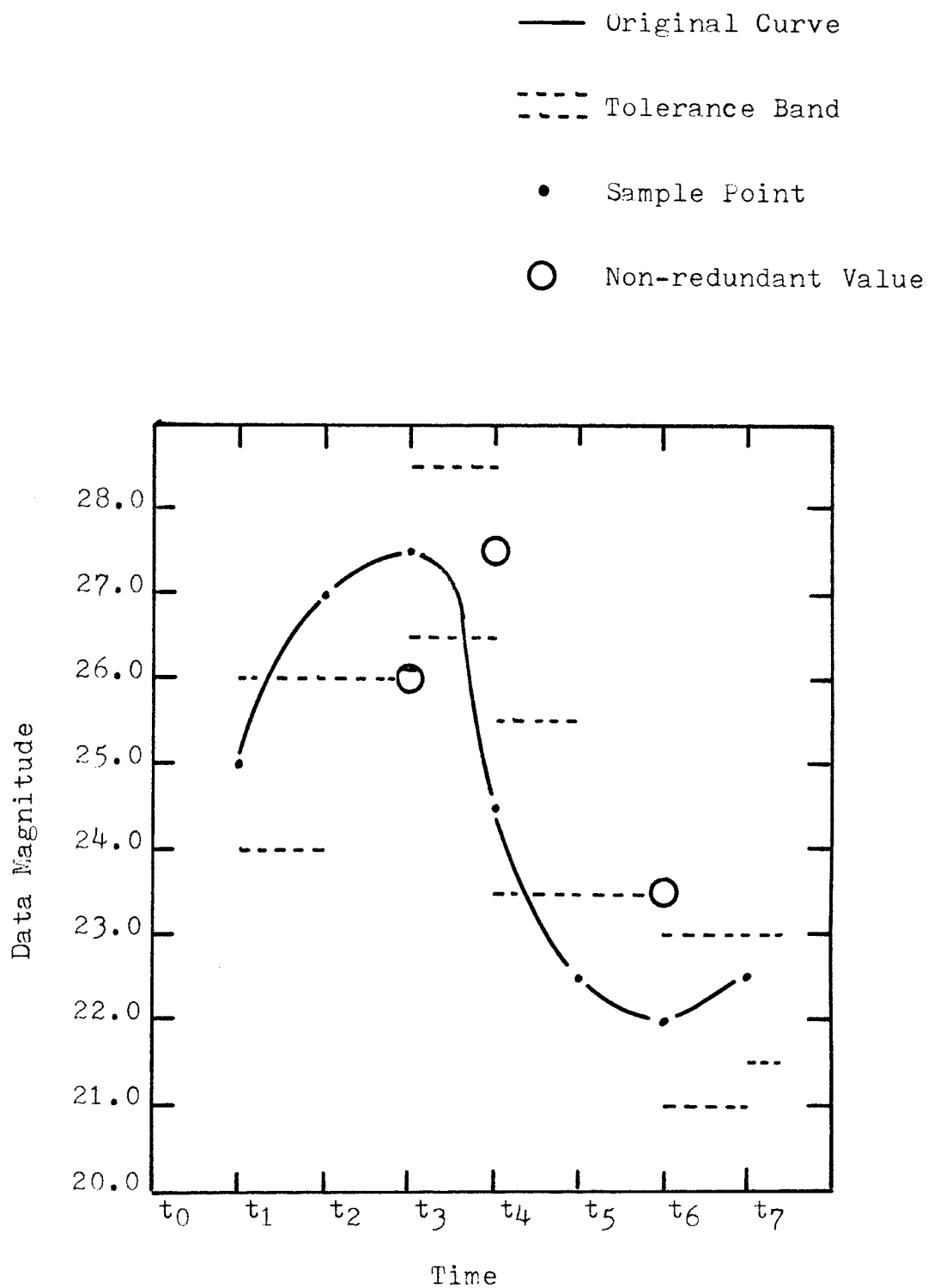


Figure 3. Zero-order Interpolator Redundancy Reduction Method

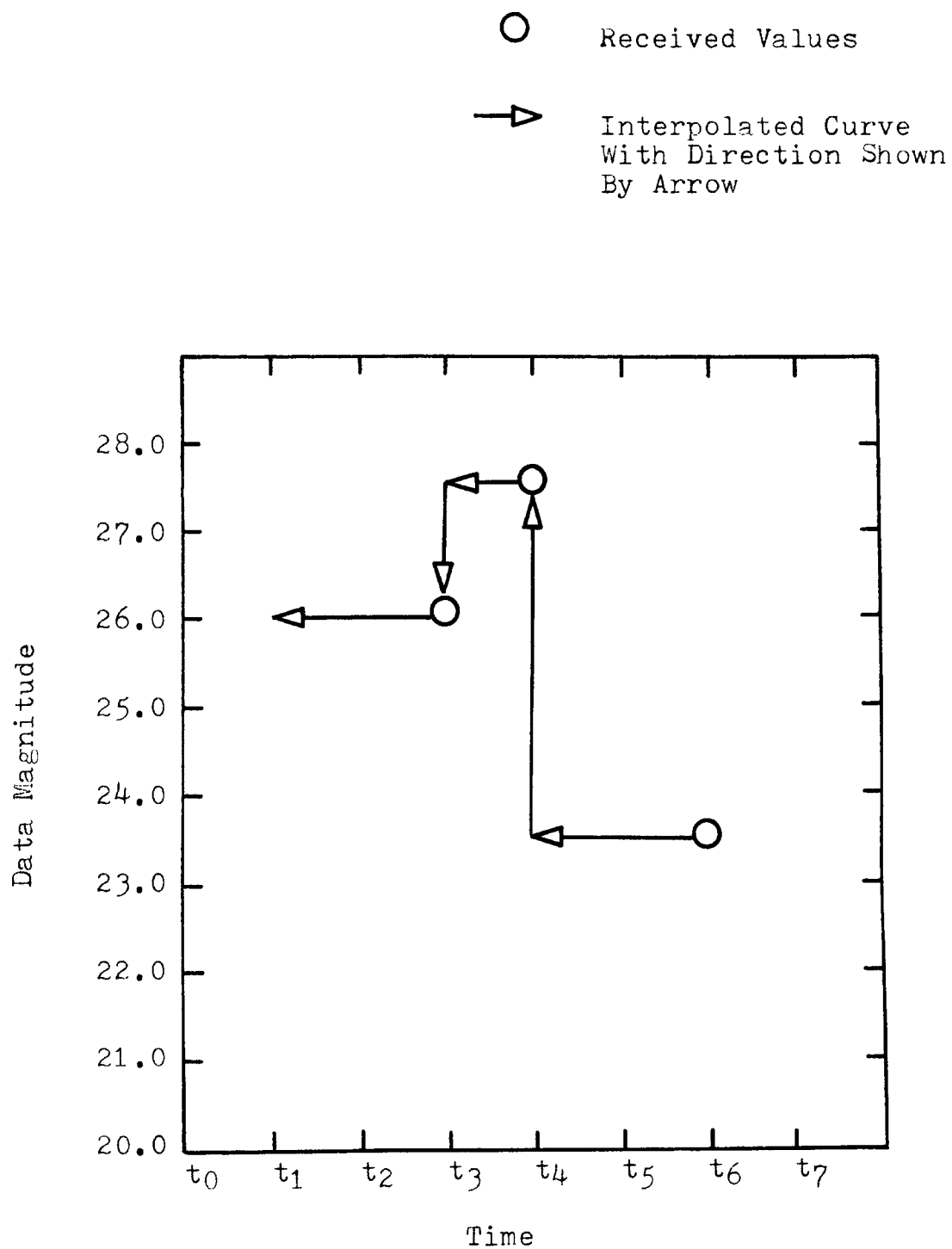


Figure 4. Zero-order Interpolator Reconstruction Technique

order interpolator which constructs the longest straight line between two sample points, such that all intermediate sample points fall within the tolerance band about this straight line. When a sample falls outside the tolerance band, the immediately preceding point is transmitted and is also used as the new starting point for another straight line. Figure 5 shows the operation of this algorithm. To reconstruct this algorithm, a line is drawn from the point received at time t_{n+1} back to the point received at t_n . The process continues in this manner as each point is received, as shown by Figure 6.

G. RECONSTRUCTION CONSIDERATIONS

Interpolators, unlike predictors, have a serious drawback when considered for real-time data reconstruction. The data may not be reconstructed at any given point in time since the value of the data at the time it goes non-redundant is not sent, but instead a value for the preceding samples is transmitted. Therefore, the proper value for data at some arbitrary point in time (t_n) is not available for use until t_{n+x} , which makes real-time reconstruction impossible.

A very often overlooked fact about data compression is that once compressed it must be reconstructed and probably decompressed for some form of time history record. The complexity of the reconstruction process must not be

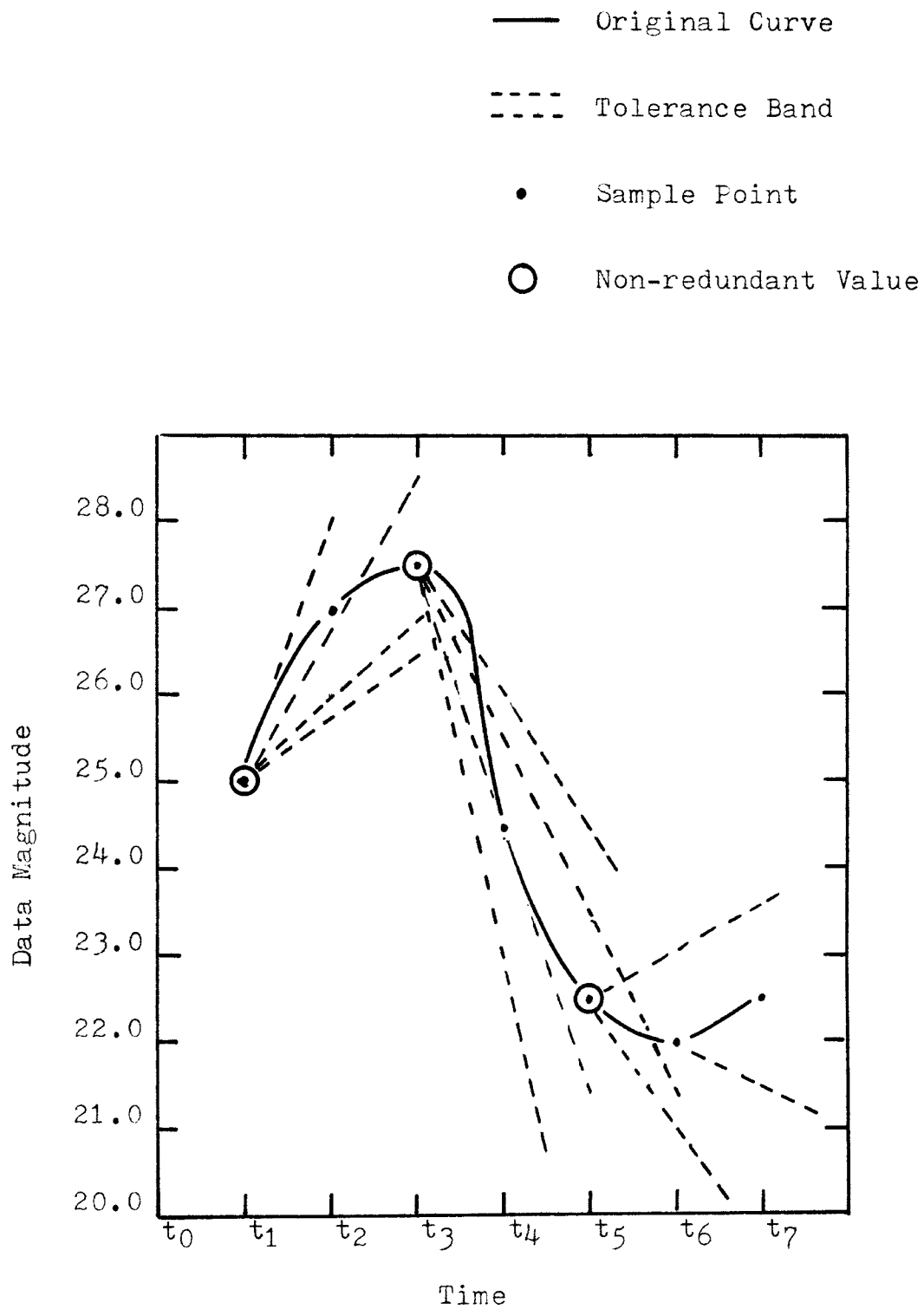


Figure 5. First-order Fan Redundancy Reduction Method

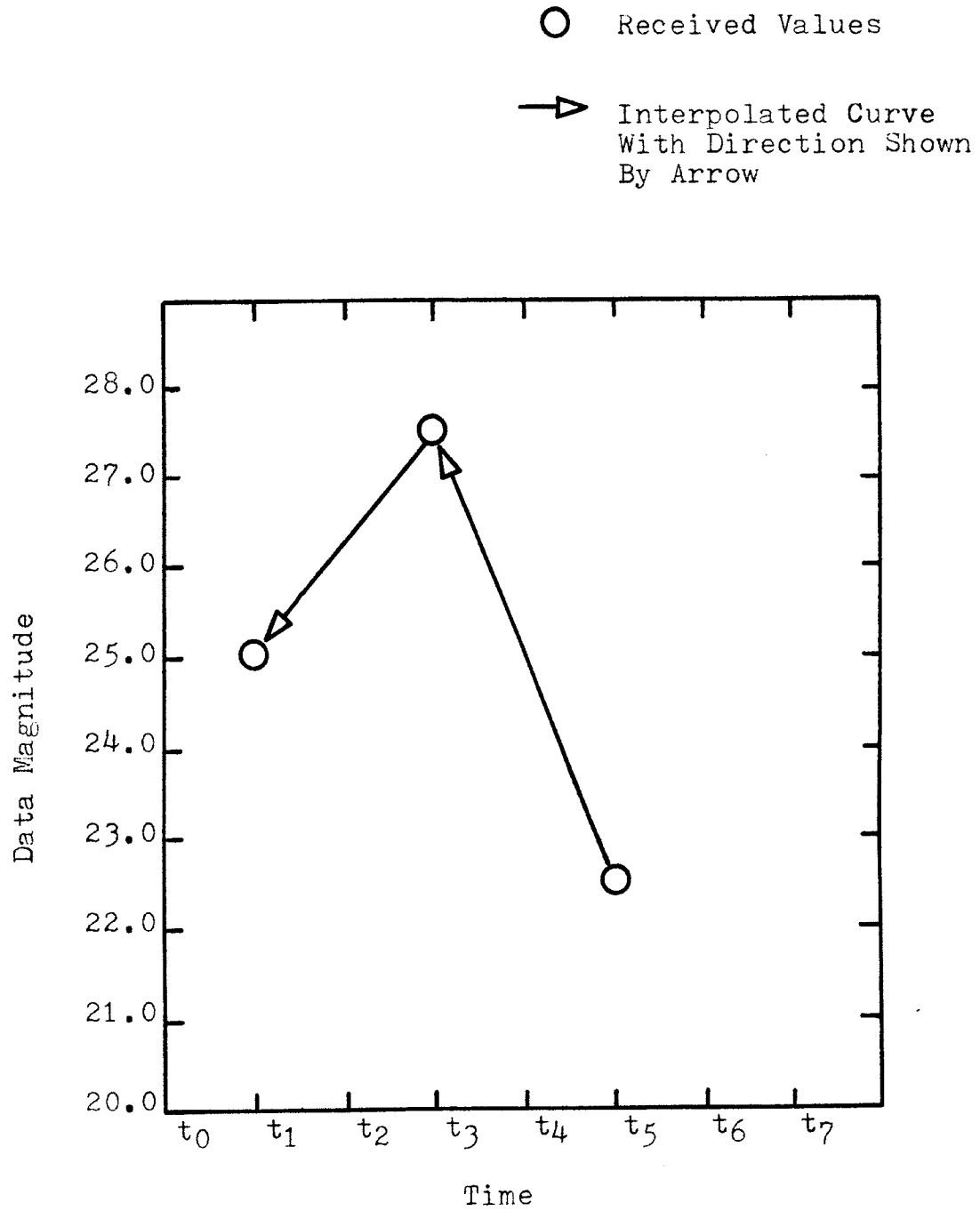


Figure 6. First-order Fan Reconstruction Technique

ignored when the choice of compression algorithms is made, since without consideration for the time and equipment required to process the data it is entirely possible that a net loss could be effected. Data compression is not a means to reduce the bandwidth requirements of an optimumly sampled system. In fact, any form of compression would have to show an increase in required bandwidth (8) over the optimum case due to the inclusion of sample identification and timing information.

III. PROBLEM CONSIDERATIONS OF REMOTE TESTING

A. BACKGROUND

Past experience has indicated that the transmission of aircraft flight data in analog form is usually not suited to voice grade telephone lines. One such experiment to determine the feasibility of using the telephone network to transmit P.D.M. (Pulse Duration Modulation) over long distance telephone lines was conducted in 1966 (10). The type of transmission available for analog signals was via a F.M. (Frequency Modulation) link. The characteristics of the proposed system were simulated using F.M. modulators, bandpass filters and F.M. demodulators. The results of this test indicated that an eight to one slowdown of input data rates would be required to keep distortion below 2%. The errors introduced and the time to transmit were already too excessive to warrant further investigation.

As a new generation of aircraft evolve, the demand for rapid data exchange between remote test sites and the central processing facility appears. The demand must be met since the development program schedules are being shortened. The current requirement is that scaled test results be available to design engineers within twenty-four hours of the flight. This demand could possibly be met by locating the necessary data processing equipment at the remote

test sites along with the engineering staff. Since engineering efforts on new aircraft cover such a wide range of disciplines as well as many laboratory support facilities, it is often impracticable to move the entire engineering staff. Locating the processing equipment at the remote site would, in most cases, result in unwanted duplication of equipment already available at the central facility. As previously mentioned, the use of a voice grade or less telephone line for transmission would be economically desirable. The present modems (modulator/demodulator), when used with a conditioned voice grade line, can be expected to transmit digital data at rates of 4,800 bits per second (1).

B. FUTURE PROJECTIONS

Data requirements for aircraft in both the commercial (11) and military fields (12) are quite extensive. A conservative estimate of an average amount of instrumentation would consist of one source of P.C.M. (Pulse Coded Modulation) and eight sources of multiplexed F.M. data. The data bandwidth required to transmit this information in real time would be in excess of 139,000 bits per second. This figure is based on: (a) the P.C.M. source having 300 words per frame, 8 bits per word and 10 frames a second for a total of 24,000 bits per second; and (b) eight sources of F.M. data with six subcarriers per source. It is assumed that the subcarriers require a sample rate of 300

samples per second and like the P.C.M., are quantized into 256 parts. The F.M. would thus require 115,200 bits per second. Combining the P.C.M. and F.M. requirements, the total is 139,200 bits per second or 17,400 words per second.

By carefully selecting only those portions of a flight which contain the significant data, it is usually possible to discard up to $2/3$ of the data collected. This is possible due to the fact that the collection of data must be started prior to the time at which the aircraft obtains the desired attitude, altitude, airspeed, etc. for the required test condition. Since the actual conditions being sought are difficult to determine in real time, there must also then be a continuation of data collection past the apparent point.

Therefore, the time required to transmit the useful data may be increased by a factor of three without causing an excessive buildup of data waiting to be transmitted. However, a reduction by a factor of ten is required to fit this data within the capacity of a voice grade line.

C. POSSIBLE SOLUTIONS

Data compression techniques employed on flight tapes could possibly provide the required ten to one reduction of effective data rates. Evaluations of the various compression techniques, including the complexity of data

reconstruction and the hardware cost/complexity of implementation, seemed to point, in agreement with others (5), (6), to two of the more simple algorithms. They are the zero-order predictor and interpolator with floating apertures.

A study (13) to determine what order was most efficient was conducted using spacecraft flight data. The results indicated that the flight telemetry data was most like zero-order curves and that the predictor provided about a 10:1 compression ratio while the interpolator provided about a 75:1 compression ratio. To verify that these algorithms would in fact provide a reduction of effective data rates on aircraft flight data, a series of studies was conducted.

Thus, it was discovered (14), (15) that the zero-order interpolator was from two to three times more effective than the zero-order predictor for most data. The zero-order interpolator was then implemented in software on a flight data pre-processing system. This test (16) predicted that approximately eighty percent of the pulse data would provide compression ratios of nine or more (see Figure 7) when using the zero-order interpolator with apertures of one percent or greater. Other studies (15) revealed compression ratios of fifty to one for analog data.

Data Source - 67 Flights

Data Coding - Pulse Duration
Modulation

Compression - Zero-order
Interpolator

Tolerance - $\pm 0.97\%$

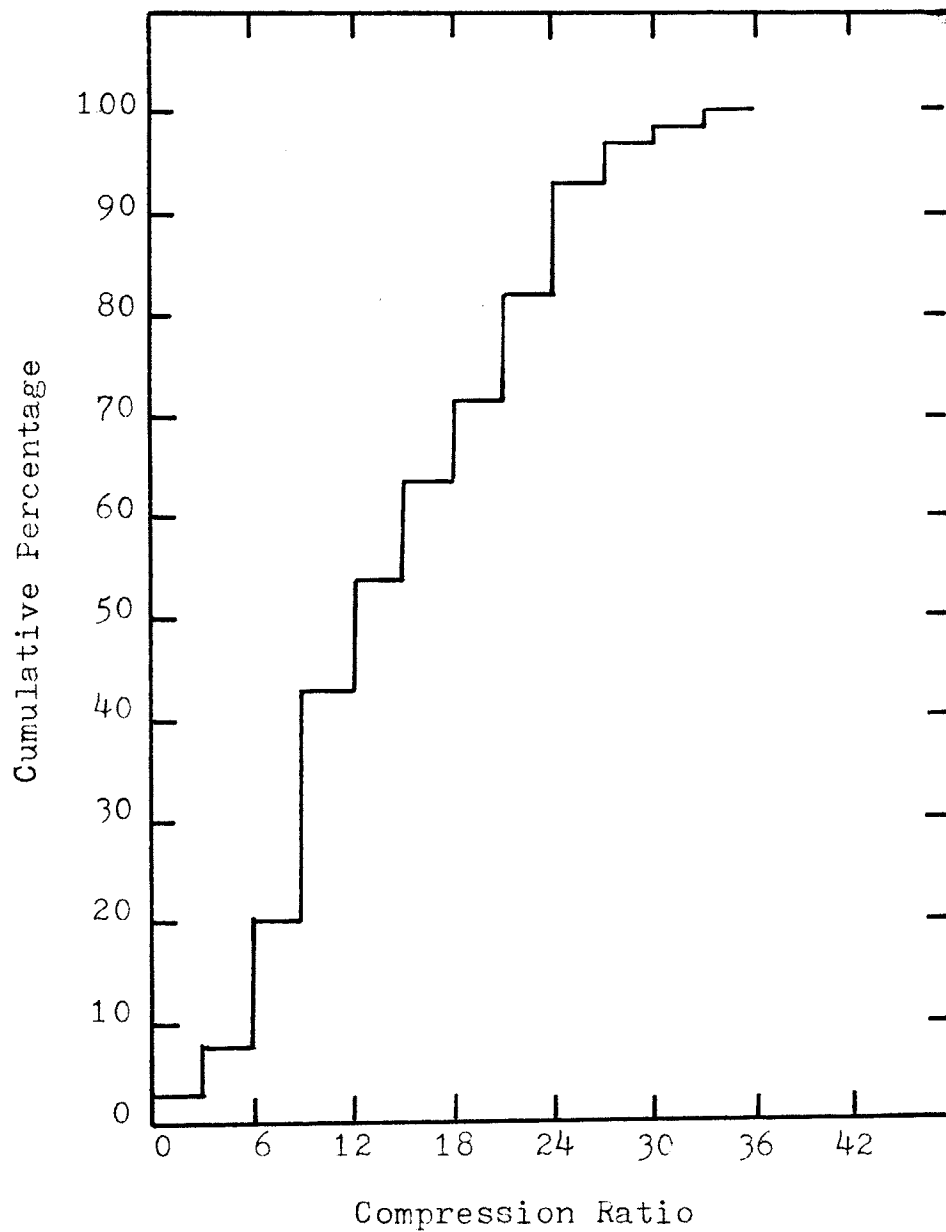


Figure 7. Compression Ratio Distribution

D. TIME AVERAGING

Usually when the subject of data compression is discussed the strengths and weaknesses of the algorithms are pointed out along with the type of data that may be well suited to a particular compression technique. One very powerful compression technique rarely mentioned is that of time averaging. Since most data sampling rates are picked for the highest expected rate at the worst case test conditions, the subsequent tests for steady state performance during steady state conditions will yield many excess sample points.

Although most compression algorithms will provide rate reductions, the inclusion of a simple time averaging algorithm will provide a very useful rate reduction technique. Many performance evaluation programs require the average values of several measurands at the same point in time to accomodate calculations which will yield other information about the test vehicle. If other compression techniques are used, the data must undergo some reconstruction process prior to the desired averaging. In addition to the rate reduction there will be fewer computations required when the averaging is done prior to transmission.

E. IDENTIFICATION

Each measurand must be accompanied by an identification tag and some sort of time position information. The

inclusion of these data will reduce the effective compression ratios in any compression scheme. Therefore, only the minimum amount of timing and identification information should be included in the output transmission.

F. BUFFER CONSIDERATIONS

Since one purpose of data compression is to reduce the effective data rates, it is reasonable to expect a relatively high speed input device and a relatively low speed output device. Since the non-redundant data rate is variable, there must be some sort of buffering scheme provided to allow the bursts of data to be temporarily stored while waiting to be transmitted. It has been demonstrated (16), (17) with flight data that a buffer length of approximately 3,000 words will be adequate for 99% of flight data with peak errors of no more than $\pm 0.5\%$. Since the input/output rate ratio of the typical situation would be 29:1, then by extrapolation, the required buffer size to provide a high degree of protection against buffer overflow would be around 9,000 words. A buffer overflow condition would cause non-redundant data to be lost and could prove to be disastrous to the test results.

Methods employing relatively small buffers (18), (19) have been developed. These utilize increased tolerance bands based on some data dependent variable. The fault common to all control methods is the possibility of

overflow even though the probability of its occurrence has been reduced.

Another approach which can be taken when the system is ground based and therefore not limited by size and space is to provide a small core buffer followed by a digital tape recorder. The tape then becomes the buffer between the variable non-redundant data rates and the bandwidth limited transmission link. It also provides some advantages which are exemplified by the ability to retransmit blocks of information if errors are encountered during transmission.

IV. DESIGN CONSIDERATIONS

Using the aforementioned studies as a basis, a practical data compression system can be evolved. The algorithms to be implemented are: zero-order, floating aperture predictor (hereafter referred to as Z.O.P.); zero-order, floating aperture interpolator (hereafter referred to as Z.O.I.); and the arithmetic average of some variable number of samples (hereafter referred to as A.V.E.).

A. ALGORITHM DESCRIPTIONS

Z.O.P.: A series of decisions and operations predict that data samples will fall within some zero-order tolerance range of a previous reference data sample. If the samples do lie within this predicted range they are considered redundant and ignored. Any data sample that is beyond the tolerance band is considered to be significant and is passed on as a non-redundant data sample. The value of this point replaces the previous reference data sample and the tolerance range relocates so as to center about the new reference point for subsequent comparisons with the following data samples. Figure 8 is a flowchart for the implementation of this algorithm.

Z.O.I.: A series of decisions and operations place a tolerance range around a reference data sample. Then, unlike Z.O.P., a similar tolerance range is placed around each subsequent data sample. If there is any overlapping

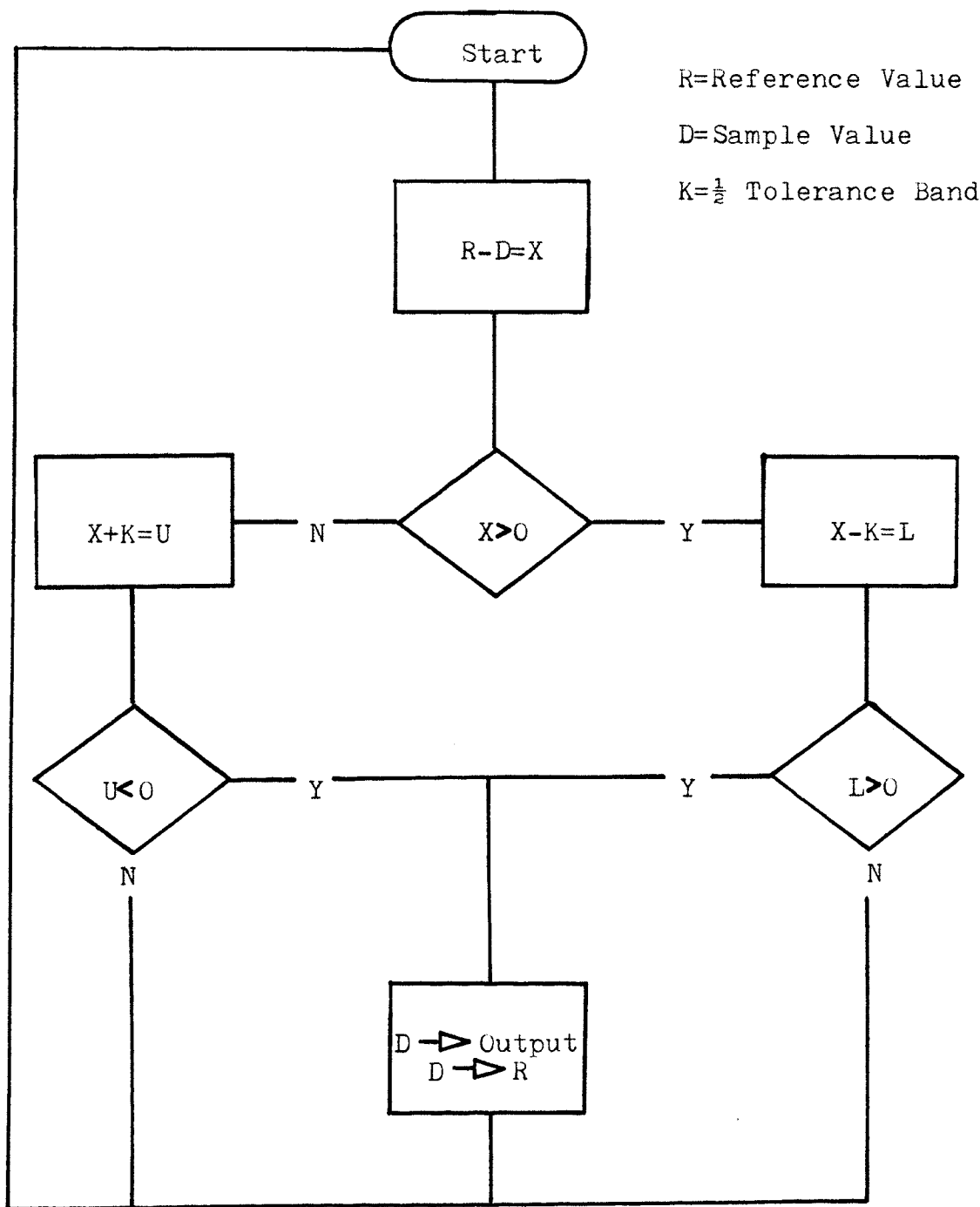


Figure 8. Flow Chart For Z.O.P.

of the tolerance ranges, the data sample is considered redundant and discarded. The original tolerance range is then reduced to consist of only the area of overlap. This process of comparing and reducing continues until the tolerance range of a data sample does not overlap the reference tolerance range. This data sample is considered non-redundant and a new reference range is established around this value. This non-redundant sample is not transmitted, but the midpoint of the previous (as modified by the process) tolerance range is sent as the significant or non-redundant data sample. As previously mentioned, data processed by use of this algorithm are more difficult to reconstruct than that using a predictor since it must be reconstructed from the most recently received point backwards in time to a previously received point. The mathematical expression (3) for the value of the non-redundant point is

$$O = \frac{U_1 + L_1}{2}$$

where O is the non-redundant point, U_1 is the most positive sample of the set and L_1 is the most negative sample of the set. Figure 9 is a flowchart for the implementation of this algorithm.

A.V.E.: A prescribed number of data samples are averaged to yield one value of that measurand for that

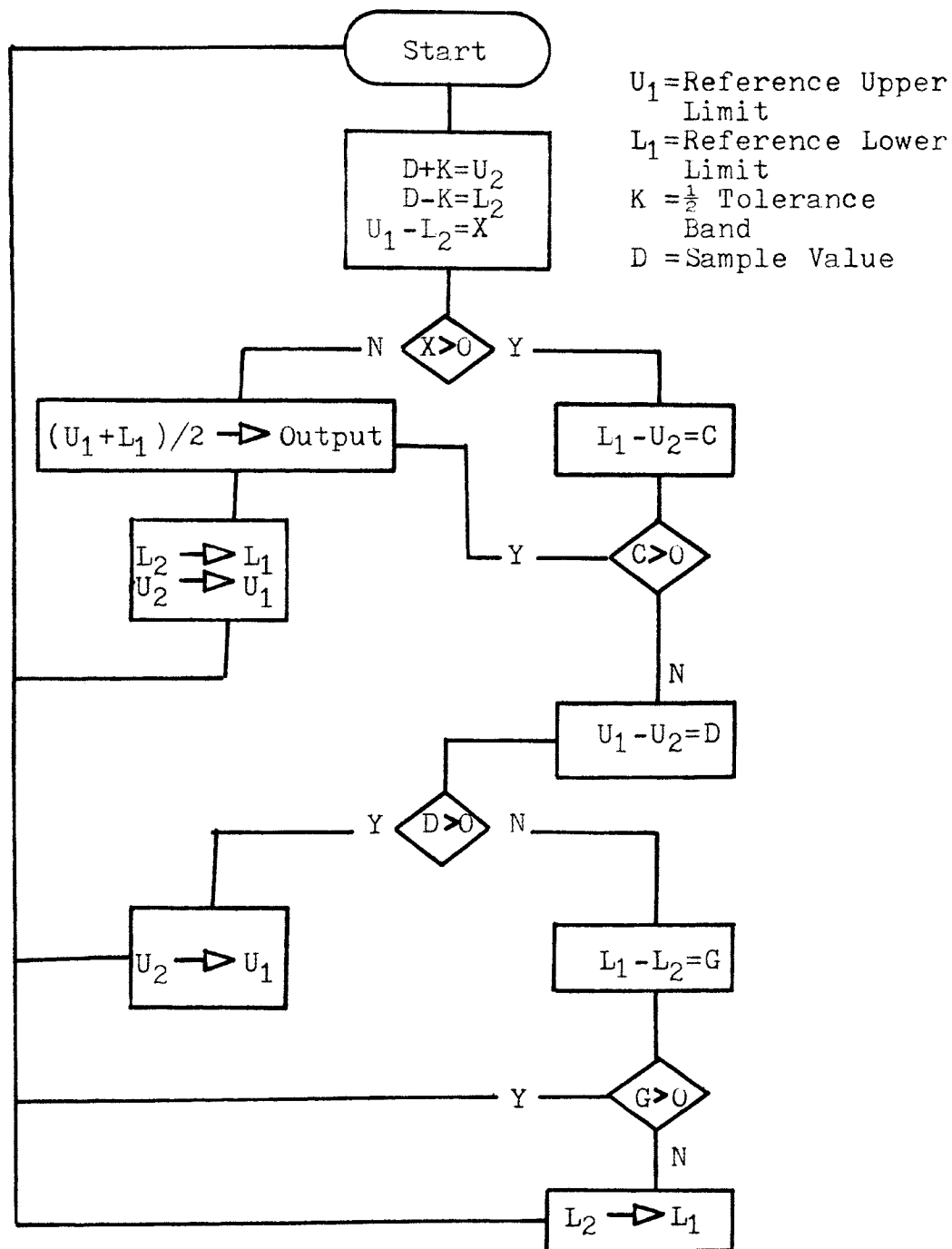


Figure 9. Flow Chart For Z.O.I.

time period. Either time or a finite number of samples may be used to determine the sum to be averaged. If a limited number of samples is used as the basis for the averaging, the division can be very simple. Thus by restricting the selection of the number of samples to be averaged to an integer power of 2, the number of samples equals 2^k . Then the proper average may be obtained by shifting to the right by k places once the proper number of samples have been taken.

B. HARDWARE COMPRESSOR V.S. SOFTWARE COMPRESSOR

There is very little doubt that the speed of hardwired processors can be made greater than that of general purpose processors. The adaptability of a software system to many different situations is its strongest point, and illustrates the hardwired system's weakest point, which is the difficulty of changing its operation. The special purpose hardwired system has a high cost in design time when contrasted to the programming cost of one of the vast number of available general purpose mini-computers. The biggest single advantage of a hardwired system is its speed.

During the testing of data compressibility (16) a software implementation of the Z.U.I. algorithm was employed. An attempt was made to evaluate four different tolerance bands on each data point. It was found during program checkout that consistent operation was impossible

because of insufficient time between data samples to run four different passes through the algorithm. When the number of tolerance bands used was dropped to three, all processing problems disappeared. Thus a boundary value for input rate versus processing time was established.

The input data rate was 900 words per second, indicating that the maximum rate was somewhere between 2,700 words per second (input rate times the number of serial algorithm passes) and 3,600 words per second. Since the cycle time of the computer (Beckman 420) was 3.2 microseconds a timing estimator is available. The approximate cycle time required of a given computer can be related to the required data word input rate by the following relationship:

$$C_a = \frac{2,700 \times 3.2}{I} \times 10^{-6}$$

where C_a = Approximate cycle time
 I = Input word rate

As mentioned in Chapter III a typical input rate is 17,000 to 18,000 words per second. To use a software approach would require a cycle time to be in the order of 500 nanoseconds ($C_a = \frac{2,700 \times 3.2}{18,000} \times 10^{-6} = 480 \times 10^{-9}$) just to meet the minimum expected rate. To allow for some input rate cushion, a safety factor of two would not be at all unreasonable. This would require a cycle time of no more

than 250 nanoseconds which is beyond the range of many current mini-computers. Therefore, a hardware approach seems to be the advisable method to meet a design goal of thirty-six kilohertz input word rate.

C. ALGORITHM MODIFICATIONS

The requirements for time position information and identification tags necessitate transmission of additional information. Since most telemetry type data is frame and position oriented, there is a great deal of a priori timing information contained in the position of the measurand. This is particularly true if the clocking source for the system is stable. To illustrate this point an example will be cited. Suppose that the parameters to be measured need to be sampled ten times a second to encompass their activity. Timing information to at least 0.10 seconds is required to accurately reconstruct the measurands after sampling. Commonly, timing information is provided once per complete sample cycle (frame). The timing information provided once per frame has limited information content after its first value if the frame rate is as accurate as the time source. The first value of time provides an absolute reference point. However, the repetitive time values every 0.10 seconds could be just as well determined by adding 0.10 seconds to the current time keying on the occurrence of any one data

channel. This is true since it is guaranteed that a new value for each data channel will be provided every 0.10 seconds. If the time position of each channel is required this may be provided by using its position within the frame for exact timing information. Channel time can be computed from the following relationship:

$$T_c = F + \frac{N}{WR}$$

where T_c = Time position for channel N
 F = Absolute frame time (seconds)
 N = Channel number
 WR = Word rate (words/second)

A method that may maximize the amount of information carried in the compressed data format is to make the identification portion of the transmitted value directly related to the measurand's position in the frame. Then use only one binary bit position within the output word format to mark the passing of a frame. If any frame should happen to be completely redundant then timing information from another source would be required. This additional timing information can be treated like data. After having transmitted the absolute time position for a reference point, any further transmissions can also be compressed. This can be accomplished by using the identification portion of the timing words to uniquely identify it as to an absolute time value as well as to what portion

of the complete time value it contains, i.e., course time - hours and minutes or fine time - seconds and milliseconds.

The method of time extraction would then be merely to recognize the absolute time information by its identification tag and to increment it by the delta frame time for each "frame marker" bit recognized in the data identification tags. If a time word appears after its initial occurrence then the implied information is that all data for a complete frame was redundant and those values for the most recently received frame may be used again for any desired calculations.

A potentially powerful compression algorithm is that of complete parameter deletion. Most test vehicles are instrumented for general test conditions, and when a specific test is being conducted there are often many parameters that are not required for the determination of test results. Failure of pickups during the testing is also a common occurrence. Therefore, the complete elimination of these unnecessary or invalid measurands would greatly enhance the overall reduction of data bandwidth required. A simple delete code may be implemented in the pre-processor to yield another contribution to the reduction of data output.

The time averaging technique is a highly profitable algorithm to employ. It has unfortunately a built-in

hazard, particularly if only one or two averaged points per run are used to describe the data. Most data systems have sporadic and sometimes periodic channel perturbations. In normal data presentations these are visually discounted as "wild points" and unless their occurrence is excessive they are ignored. In the case of an averaged point, one or two of these points will have the effect of a slight biasing of the data value. This can cause major problems since the value will be close to the expected value and, therefore, not subjected to the "wild point" visual elimination.

To preclude this slight biasing of the data values while using the time averaging algorithm, a "wild point" elimination technique can be employed. This consists of accepting the first data value as a reference point and performing a modified Z.O.P. technique on all subsequent data values during that averaging period. The modification is that if the next data point is more than some tolerance band away from the previous sample, the new sample is ignored and the previous sample is added into the summation. If the new value is within the tolerance band it becomes the new reference value and also is added into the summation. This has the effect of eliminating any data sample that is more than the tolerance band away from the preceding sample. Two faults to this technique may be cited, but their price is small compared to their overall gain. One is that the first point may be a "wild

point". This would cause the elimination of all of the valid data points for that averaging period. This, however, will produce in the end product a "wild point" and, therefore, can easily be noted and ignored. The second fault is that of a valid data point being eliminated from the average because it was more than some pre-assigned distance from the preceding sample. This means that the data in question should probably not be subjected to the time averaging technique and, therefore, the answer is non-valid, with or without the inclusion of this data point. However, since any point elimination could be an indication of a problem that requires immediate action (noisy pick-up, data not amenable to time averaging) the use of a flag in the final data word to indicate the exclusion of one or more data samples will serve to alert the final processing system that there is some unresolved question about this data point.

The supercommutation (multiple occurrence within a frame) of a measurand can pose a problem to data compression techniques. Since the ultimate goal is to remove as much redundant data as possible the supercommutated measurand should not be considered as several different parameters. This is because to properly implement compression, the two most recent successive samples of a measurand must be used for comparison. The use of a stored decommutation structure can be used to point one or more measurands to

the same control area. This will provide the proper order of comparison. A remaining point to consider is that of ensuring that each position has a unique identifying tag. This ensures that upon reconstruction the non-redundant value will be properly placed.

Another facet that should be considered about supercommutation is that only one occurrence of a measurand per frame should be included in a time averaging period to preclude unnecessary complications in the averaging technique.

Another possibility which can occur is that of a measurand being subcommutated. That is, one word position in the frame may be used for different measurands on subsequent frames until some number of frames have occurred and then the first measurand reappears. The most direct method is to compress only on the word position without regard to the current measurand. However, there is no reason to assume that any two successive subcommutated parameters will be within a tolerance band of each other. It also precludes the probable high compression ratios obtainable on these measurands. The expected activity of the measurands must be low or they would not have been relegated to a subcommutation position. Again, the use of the stored decommutation structure lends itself to this situation for a ready solution. This time, instead of the pointer directly pointing to the control area it can point

indirectly by the use of a frame level register to find the control area. This also provides a method to uniquely identify each measurand and to delete unused positions on the subcommutator.

When the problems associated with the reconstruction of compressed data are examined one point becomes apparent. Vast amounts of storage are required to reconstruct the data if worst case conditions are encountered. It is possible to get only one (or zero) points from the entire test on any number of measurands since only the non-redundancy of a point following the first sample can cause a transmitted value. In the case of Z.O.I. the reconstruction is done in the reverse order which carries with it the possibility of having to store a complete test before the reconstruction process could start.

To preclude this possibility the algorithms can be altered to provide a forced data readout at the first frame of data in a run, every n frames, and a forced frame readout at the end of a run. Another benefit of this forced redundancy is that the system's performance can be continually evaluated during the reconstruction process, thereby providing a method to detect system malfunctions.

D. USE OF A GENERAL PURPOSE COMPUTER FOR CONTROL

Once having decided to implement the algorithms in hardware the next step is to determine how to get the

necessary control information into the hardware system and how to effectively change it, from run to run and test vehicle to test vehicle.

A great deal of the information required is measurand oriented. The use of a punched card as the vehicle to carry the measurand information is a natural choice. This allows for individual changes to be easily made. Since different test vehicles will have a wide variety of instrumentation set-ups the ability to alter the set-up within the compressor is a certain requirement.

The method of control for the selection of which portions of a test to process requires a device that can input the reference data, compare it with programmed values, and cause the hardware compression system to start and stop.

A small general purpose computer can be quite effective in the handling of the above mentioned duties in addition to being a perfect device for the small core buffer followed by a digital tape recorder. The input/output structure of many current mini-computers is quite adequate to handle the required control functions.

V. IMPLEMENTATION

A. BASIC COMPRESSOR

The compressor is a special purpose computer, responsive to a limited number of macro-instructions. This design philosophy was chosen for the speed of operations that it offers. Thus, a high input word rate may be accommodated with a resultant low average output rate. The compression algorithm (excepting deleting) which is used on the data for a particular run is common to all data, with the provision to alter this choice for subsequent runs. Figure 10 illustrates a block diagram of the entire system. As each digital word enters through the digital multiplexer input, an address for the associated memory is generated. This address is used to call a unique word from a resident decommutation list. There are as many entries in this list as there are measurands in the input data stream. Thus, each input word (measurand) generates an address which calls from memory a word whose contents are used for control and subsequent direct and indirect addressing. Bits 1 through 12 of this accessed word contain the address of another word in memory assigned to the measurand in question. Current measurand values (which are necessary to perform the assigned algorithm) are thus preserved in an addressable location.

If the data is supercommutated, then all list words

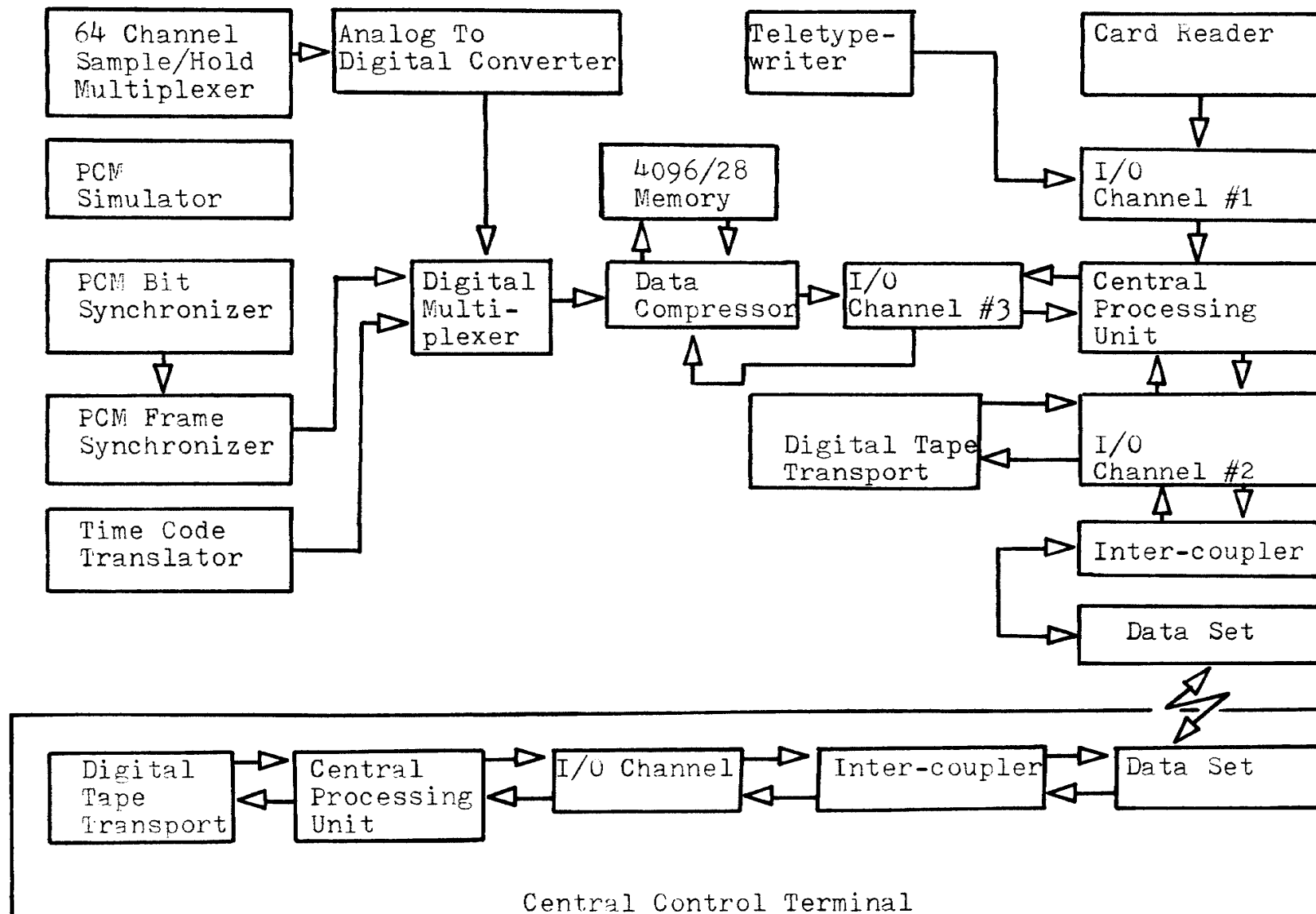


Figure 10. System Block Diagram

associated with a measurand contain the same address in the control word (C.W.) field. This makes sure that the most recent information is always used during the compression algorithm. If it is necessary to output this measurand, an output address is generated which has a correspondence to the time position as well as the identity of the measurand. In the case of subcommutation, the C.W. field is a base address. To form the true C.W. address, the base address is added to the current value of the subcommutation level. For data at the main frame rate, the address field is used to access the proper control word for further processing.

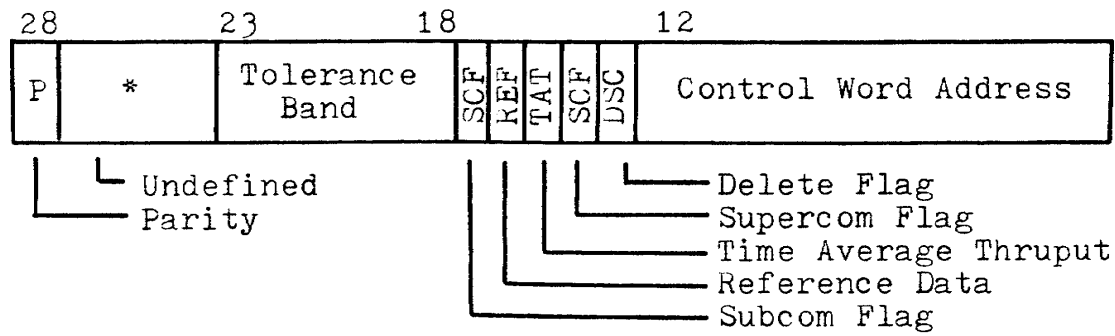
B. ANALOG SUBSYSTEM

The analog subsystem is made up of sample and hold amplifiers, a multiplexer and an analog to digital converter. The sample rate is determined by using a submultiple of the P.C.M. word rate. This provides correlation between the P.C.M. and the analog data. Since the input has sixty-four sample and hold amplifiers sixty-four addresses are required for the multiplexer. The method chosen to generate these addresses was to build a list in the memory that can have up to sixty-four entries. These words contain the multiplexer addresses as well as control flags. This provides the ability to address all inputs one time or to hold multiples of shorter scan cycles.

For ease of discussion, one cycle through this list will be called an analog scan cycle. The sample pulse initiates a sample of all input amplifiers. The analog list is then accessed. Each word in the list contains an address for the multiplexer. When an "end of scan" control flag is found in one of the list words the cycle stops. When a "recycle" flag is encountered, the cycle again stops and resets the address generator to the first address in the list. Thus by proper placement of "end of scan" and "recycle" flags the analog scan cycle may be made to encompass from one to sixty-four list words. These words in turn may each contain a different multiplexer address. Alternatively there may be repeated address structures between the "end of scan" flags to accommodate supercommutation for the high frequency analog inputs. Figure 11 illustrates the analog list word structure (20).

The analog words are normalized to percent of the source full scale in the compressor prior to performing any algorithms. This is accomplished by first storing within the compressor memory offset and scale values during a pre-run initialization pass. Then during run time the offset is subtracted from the data word value and the result is multiplied by the scale in a hardware arithmetic unit. The final resultant normalized value is then subjected to the algorithms.

P.C.M. List Word



Analog List Word

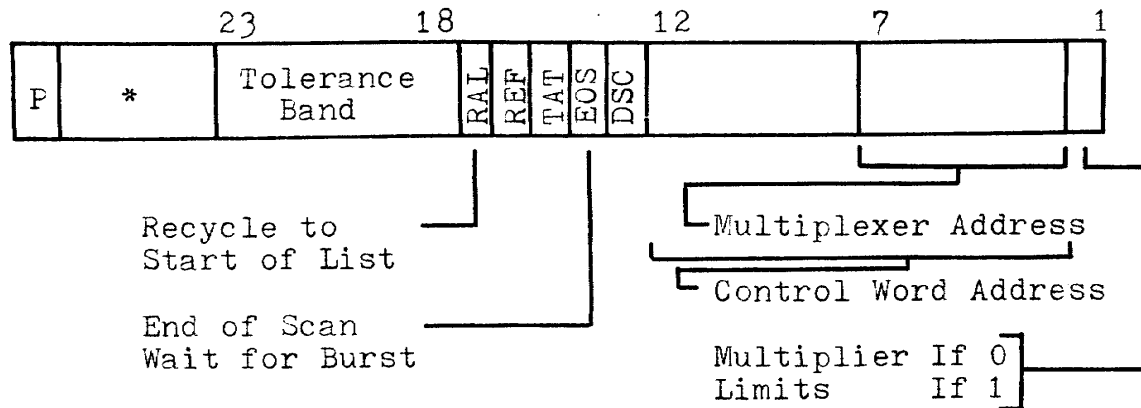


Figure 11. List Words

C. LIST WORD

The basis of all subsequent processing is in the contents of the list word. As each digital word is multiplexed into the system an address is generated corresponding to the time position of the input word. Thus, the address is related to the word position. The address is used to obtain the list word from the memory which contains processing information unique to the measurand occupying that time word position. Figure 11 depicts the P.C.M. and Analog List Words.

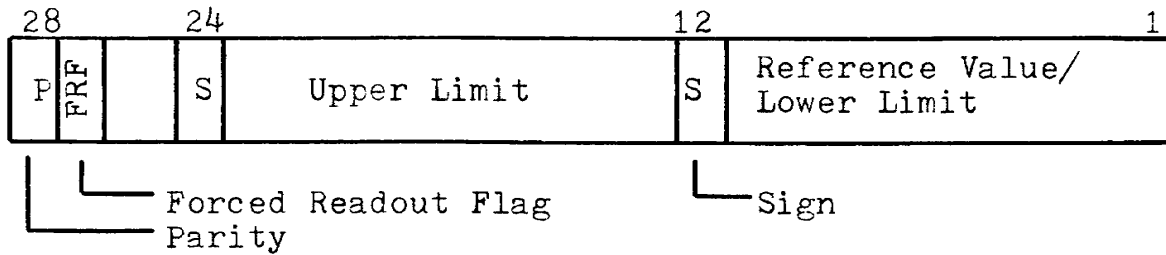
D. CONTROL WORD

The control word has three forms. The form used depends on the algorithm and the data source. If the data source is analog, two control words are associated with each analog list word. The first contains the necessary information (zero offset and scale factor) to normalize the input value, while the second is identical to the control word used for P.C.M. The third variation is associated with the A.V.E. algorithm. This control word is used to hold the current reference value as well as the accumulating sum. Figure 12 depicts the three variations (20).

E. INPUT INFORMATION

The general purpose computer obtains the structure

Z.O.P./Z.O.I. Control Word



Analog Multiplier



A.V.E. Control Word

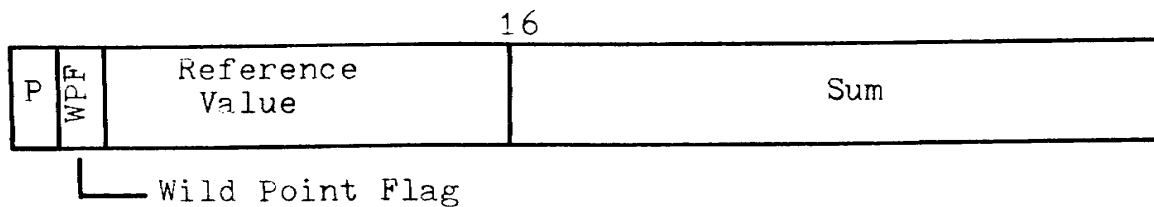


Figure 12. Control Words

of the P.C.M. format from punched cards. These cards contain the names and locations of measurands in the P.C.M. format. Information relative to the applicability of specific tests is also provided. Thus, sufficient information for compilation of the required decommutation list for the compressor is available.

F. EDITING

The first gross removal of redundant data may be accomplished through the selection for processing only those portions of the flight that contain significant data. The approximate times of the areas may be noted by the engineering personnel at the flight monitoring stations.

These noted times may then be programmed into the general purpose computer for start/stop control. The timing signal during playback is being supplied to the computer from the time track on the flight tape via time code translation equipment.

G. CONTROL

The control of the compressor is exercised by the general purpose computer which through time or reference information (run number, program, etc.) can determine what areas of the tape should be processed based on previously furnished information.

Since each run may require different tolerance bands, various deletion changes and algorithm selection, a modification of the compressor core may be required. This is accomplished through a two way high speed I/O channel. Between runs the computer re-configures the compressor core via this channel. During a run the compressor uses this channel to transfer the non-redundant data to the computer for subsequent transfer to digital tape.

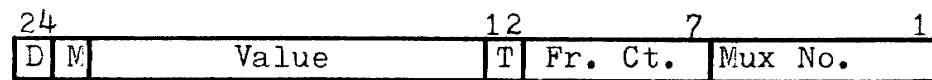
H. DATA IDENTIFICATION

The identification of the measurands is provided by using a portion of the memory address of each list word. This provides a correspondence between identification and P.C.M. word number and F.M. sample position. The final processing computer then may associate the identification directly to a list containing the measurand's name. Figure 13 illustrates the output word format (20).

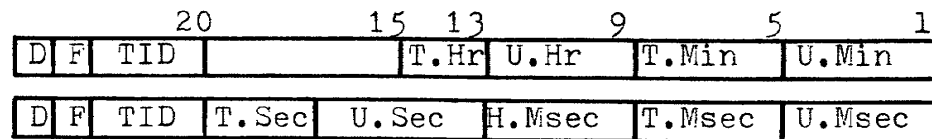
The identification of the analog words is provided by using the low order bits of the address of the list words concatenated with a counter which increments for each recycle of the analog list. This counter is reset at the P.C.M. frame rate. This provides unique addresses to identify the input both in measurand identity and time position. Using this method the correlation of the two sources is maintained since the location of the initiate or "start of scan" pulses in the P.C.M. frame is known.

24		12		1	
D	M	Value	T	Channel ID	

Analog



Time



D - 1=Data	Fr. Ct. And Mux No.=Channel ID
0=Time	Channel ID=Unique Address
M - 1=First Transmitted	Associated With One
Sample Of A Frame	Measurand
(Time Marker) Or	
Wild Point Flag	T - 1=Pulse
In A.V.E.	0=Analog
	TID - 0,6,7=Not Defined
	1=Forced Time
	2=Normal Time
	3=End of Average
	4=Start of Average
	5=Rate Overflow

47

I. DATA TRANSMISSION

Data transmission is accomplished using the general purpose computer for the control of a digital tape unit and a modum (modulator/demodulator) interface. A receiving system is at the central collection location for the reception of the data. A duplicate digital tape is the end product of a successful transmission.

This "off line" approach is advantageous since it is very inefficient to make multiple restarts of the processing to accommodate error recovery procedures. These procedures are necessary to employ when telephone line problems cause transmission errors. It is important that the transmission system is provided some means of error detection and recovery since only the non-redundant data values are being transmitted. A reasonable error protection is that of the I.B.M. cyclic redundancy check character. This provides for the detection of up to three bit errors and all odd numbers of errors over three (21) within the transmitted block. The recovery procedure that will be employed is that of retransmissions of the record that gave the error indication.

VI. CONCLUSIONS

An extensive testing program can be conducted at a remote site using only a voice grade telephone line to transmit the data to a central facility for subsequent processing.

The restrictive bandwidth of the voice grade telephone line is made usable through data compression and editing techniques. The combination of these two methods will allow data sample rates of 139,000 bits per second to be reduced to an effective rate of less than 4,800 bits per second.

These reductions may be realized only through the appropriate choice of compression technique used on the input data. Therefore, the order or type of algorithm must fit the data or an increase of required bandwidth may result.

Once the proper method of compression has been established other savings may be found by using characteristics of the data to provide part of the required timing information. Additional savings will be obtained by using "selective channel stripping".

Careful choice of those areas of the test to transmit, and equally thoughtful selection of compression aperture size combined with the above mentioned techniques will provide bandwidth reductions of approximately 30:1.

BIBLIOGRAPHY

1. Bhushan, Abhay K. "A Survey of Data Communication Devices and Facilities," Computer Design, pp. 56-65, July, 1969.
2. Kortman, C. M. "Redundancy Reduction - A Practical Method of Data Compression," Proceedings Of The IEEE, Vol. 55, No. 3, pp. 253-263, March, 1967.
3. Andrews, C. A., Davies, J. M., Schwarz, G. R. "Adaptive Data Compression," Proceedings Of The IEEE, Vol. 55, No. 3, pp. 267-277, March, 1967.
4. Kortman, C. M. "Data Compression And Adaptive Telemetry," Wescon/65 Record, 14.4, 8 p., August, 1965.
5. Meigs, Bruce E., Stine, Larry L. "Real-Time Compression And Transmission Of Apollo Telemetry Data," AIAA, Paper 69-970, 9 p., September, 1969.
6. Sheldahl, Stephen A. "Comparison Of Hardware Requirements For Polynomial Data Compressors," Proceedings National Telemetry Conference, pp. 83-89, April, 1968.
7. Klote, Layton A. B., Duffin, Patrick M. "Remote Site Data Compression," Proceedings National Telemetry Conference, pp. 96-102, April, 1968.
8. Gardenhire, Lawrence W. "Redundancy Reduction - The Key To Adaptive Telemetry," Proceedings National Telemetry Conference, 16 p., June, 1964.
9. Ehrman, L. "Analysis Of Some Redundancy Removal Bandwidth Compression Techniques," Proceedings Of The IEEE, Vol. 55, No. 3, pp. 278-287, March, 1967.
10. Pittman, Kenneth C. "Feasibility Of P.D.M. Transmission From Edwards Air Force Base To St. Louis Via Telephone WATS Lines," McDonnell Aircraft Company Memo 289-1271, 3 p., October, 1966.
11. Strom, J. A., Crowley, L. D. "Integrated Flight Test Data System With Real-Time Graphic Display," Douglas Aircraft Company Paper 5636, p. 2, August, 1969.

12. FTD Instrumentation and Data Departments. "F-15 Aircraft (MCAIR Model 199) Flight Test Integrated Data System," McDonnell Aircraft Company Report H277, 6.0-6.23, 14 July, 1969.
13. Jory, H. M. "Data Compression By Redundancy Reduction" McDonnell Aircraft Company Report F532, p. 2.16, 30 June, 1967.
14. Jory, H. M. "Comparison of ZFN and ZVA Techniques on PDM Data," McDonnell Aircraft Company Memo 289-1381, 3 p., November, 1967.
15. Jory, H. M. "Preliminary Results of Airplane Redundancy Reduction Investigation," McDonnell Aircraft Company Memo 289-1377, 10p., October, 1967.
16. Pittman, Kenneth C. "ZVA Compression Study Interim Results," McDonnell Aircraft Company Memo 289-1418, 11 p., July, 1968.
17. Jory, H. M. "Buffer Considerations and Queue Lengths in Airplane Data," McDonnell Aircraft Company Memo 289-1394, 13 p., February, 1968.
18. Stevens, W. A., Pownall, L. A. "Digital Data Compressor Buffer Queue Control," EASCON'68 Convention Record (1968), pp. 325-332.
19. Simpson, Richard S. "Buffer Control In Data Compression Systems For Non-Stationary Data," Proceedings National Telemetry Conference, 6 p., June, 1964.
20. Krause, Bruce H. "Flight Test Data Compression and Transmission System Specifications and Operating Characteristics," McDonnell Aircraft Company Report DSE-025, 80 p., August, 1970.
21. Cullen, John W. "Binary Synchronous Communications," Computer Design, pp. 40-45, October 1968.

VITA

Kenneth Carlyle Pittman was born in Shelbyville, Illinois on June 26, 1934. He attended public schools in Shelbyville and Vandalia, Illinois. He was graduated from Vandalia Community High School in June, 1952. He entered the University of Illinois in Urbana, Illinois, receiving a Bachelor of Science Degree in Electrical Engineering in June, 1956.

He is employed by McDonnell Aircraft Company and has been active in the utilization of digital and analog computers, particularly in applications for the reduction of flight test data. His specific duties are those of supervision of a group engaged in the design and fabrication of data reduction equipment.